8

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/778,076	02/07/2001	Yutaka Haga	1046.1236/JDH	4573

21171 7590 11/19/2004

STAAS & HALSEY LLP
SUITE 700
1201 NEW YORK AVENUE, N.W.
WASHINGTON, DC 20005

EXAMINER
YIGDALL, MICHAEL J

ART UNIT	PAPER NUMBER
2122	

DATE MAILED: 11/19/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

8

Office Action Summary	Application No.	Applicant(s)
	09/778,076	HAGA, YUTAKA
	Examiner	Art Unit
	Michael J. Yigdall	2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _30 July 2004_.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _8-17,19-28 and 30-42_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _8-17,19-28 and 30-42_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant's response and amendment filed on July 30, 2004 has been fully considered. Claims 8-17, 19-28 and 30-42 are now pending.

Response to Arguments

2. Applicant's arguments have been fully considered but they are not persuasive.

3. Applicant contends that Smolders does not discuss identifying a type of branch instruction (Applicant's remarks, page 12).

However, Smolders discloses identifying branch instructions (see column 4, lines 1-6), so as to generate an interrupt after each branch (see column 4, lines 21-22, and step 30 in FIG. 3). Smolders further discloses saving the address of the next basic block of code (see column 4, lines 31-32), so as to provide a return address for the interrupt handler (see steps 34 and 64 in FIG. 3). In other words, Smolders determines the address of the next basic block of code to return to after the branch instruction is executed. To determine this address, Smolders must identify whether the branch is taken or not taken, which is to say that Smolders must distinguish between conditional branches and other types of unconditional branches, which is to say that Smolders must identify the type of branch. Without such identification, the address of the next basic block of code cannot be determined, and the interrupt handler, as disclosed, would become inoperative. Therefore, Smolders inherently identifies the type of branch instruction.

4. Applicant alleges that no support has been provided for the argument that instruction codes are inherently decoded in order to distinguish between branch instructions and non-branch instructions (Applicant's remarks, page 12).

However, in addition to identifying branch instructions (as presented above), Smolders further discloses an instruction flow unit that dispatches instructions to selected execution units for execution (see column 3, lines 10-13). The execution units include fixed-point execution units, load/store execution units, and floating-point execution units (see column 3, lines 13-15). The instruction flow unit cannot dispatch instructions in this manner without first interpreting or "decoding" the instruction codes, so as to determine the appropriate execution unit. Furthermore, it is understood in the art that such instruction decoding is an integral part of the instruction cycle. Thus, Smolders inherently decodes the instruction codes.

5. Applicant submits that there is no motivation to combine the art because such a combination would itself require a modification of program code (Applicant's remarks, page 13).

However, Alexander discloses an apparatus for collecting a profile of a subroutine included in a program (see the title and abstract). Alexander discloses generating timer interrupts (see column 4, lines 20-23), and further suggests generating other interrupts instead of timer interrupts (see column 11, lines 22-25).

Similarly, Smolders discloses a system for collecting a trace of a program (see the title and abstract). Smolders discloses generating interrupts after every branch instruction (see column 3, lines 58-61), and further discloses that tracing by way of such interrupts is performed "without introducing any overhead or modifying the code" (see column 1, lines 64-67). The feature of not modifying the code is in contrast to the undesirable method of inserting additional instrumentation code (see column 1, lines 39-46).

One of ordinary skill in the art would have been motivated to enable tracing without introducing any overhead and without modifying the code, as taught by Smolders. Therefore, it

would have been obvious to one of ordinary skill in the art at the time the invention was made to substitute the timer interrupt of Alexander with the branch interrupt of Smolders, as suggested by Alexander, so as to preclude any overhead and modifications to the code.

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 8-17, 19-28 and 30-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,002,872 to Alexander, III et al. (art of record; herein "Alexander") in view of U.S. Pat. No. 6,253,338 to Smolders (art of record; herein "Smolders").

With respect to claim 12 (currently amended), Alexander discloses an apparatus for collecting a profile of a subroutine included in a program (see the title and abstract), comprising:

(a) a storage unit storing a profile (see FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles; note that each node is considered a storage unit);

(b) an analyzing section, when an interrupt is generated during execution of said program, obtaining a branch source address and a branch destination address from a source of said interrupt (see column 5, lines 20-32, which shows analyzing the stack frames in response to an interrupt to identify subroutines, and column 5, lines 41-62, which shows obtaining a call or branch source address and a return or branch destination address).

Although Alexander discloses generating timer interrupts (see column 4, lines 20-23), and suggests that other interrupts may be generated instead (see column 11, lines 22-25), Alexander does not expressly disclose the limitation wherein the interrupt is generated by execution of a branch instruction, and Alexander does not expressly disclose identifying a type of said branch instruction by obtaining an instruction code from said branch source address and decoding said instruction code.

However, Smolders discloses generating an interrupt by execution of a branch instruction (see column 3, lines 58-61). Smolders further discloses identifying branch instructions (see column 4, lines 1-6) and identifying the address of the next basic block of code after the branch instruction (see column 4, lines 31-32), so as to provide a return address for the interrupt handler (see steps 34 and 64 in FIG. 3). To determine this address, Smolders must identify whether the branch is taken or not taken, which is to say that Smolders must distinguish between conditional branches and other types of unconditional branches, which is to say that Smolders must identify the type of branch. Without such identification, the address of the next basic block of code cannot be determined, and the interrupt handler, as disclosed, would become inoperative.

Smolders further discloses an instruction flow unit that dispatches instructions to selected execution units for execution (see column 3, lines 10-13). The execution units include fixed-point execution units, load/store execution units, and floating-point execution units (see column 3, lines 13-15). The instruction flow unit cannot dispatch instructions in this manner without first interpreting or "decoding" the instruction codes, so as to determine the appropriate execution unit. Furthermore, it is understood in the art that such instruction decoding is an integral part of the instruction cycle.

Therefore, Smolders discloses a system for collecting a trace of a program (see the title and abstract), wherein an interrupt is generated by execution of a branch instruction, and wherein the type of the branch instruction is inherently identified by obtaining an instruction code and decoding the instruction code. The system enables tracing without introducing any overhead and without modifying the code (see column 1, lines 64-67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the system of Alexander with the features taught by Smolders, including substituting the timer interrupt of Alexander with the branch interrupt of Smolders, as suggested by Alexander, so as to preclude any overhead and modifications to the code.

Alexander in view of Smolders further discloses:

(c) a collecting section obtaining said branch source address, said branch destination address, and a identified result from said analyzing section when the identified instruction is a calling instruction or a return instruction of said subroutine (see Alexander, column 5, lines 41-62, which shows obtaining a call or branch source address and a return or branch destination address for a subroutine); when said identified result is said calling instruction, storing said branch destination address as a subroutine address corresponding to said calling instruction and a calling time of said subroutine corresponding to said calling instruction in said storage unit (see Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a subroutine address and a base time); and when said identified result is said return instruction, obtaining a return time of said subroutine corresponding to said return instruction, calculating a execution time of said subroutine based on said obtained return time and said calling time, and storing a cumulative value of said execution time as said profile in

correspondence with said branch destination address in said storage unit (see Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including an execution time and a cumulative time, inherently calculated based on the calling time and the return time; see also FIG. 4A, which shows timestamps for entering and returning from subroutines).

With respect to claim 8 (currently amended), Alexander in view of Smolders further discloses the limitation wherein a plurality of storage units respectively corresponding to a plurality of executors of said subroutine are prepared (see Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a plurality of nodes or storage units); and

said collecting section specifies said executor of said subroutine and stores said profile of said subroutine corresponding to said specified executor in said storage unit (see Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles; note the parent node pointers, which specify the executors of subroutines).

With respect to claim 9 (original), Alexander in view of Smolders further discloses the limitation wherein said collecting section individually stores profiles of a plurality of subroutines corresponding to a specified executor in said storage unit (see Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles; note that a path in the tree represents the subroutines executed by a specific executor).

With respect to claim 10 (original), Alexander in view of Smolders further discloses the limitation wherein said collecting section individually stores a first profile of a subroutine called

by a main routine, and a second profile of the subroutine called by another subroutine, in said storage unit (see Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles; see also FIG. 10 and column 8, lines 24-40, which shows individual profiles of subroutines organized based on the calling routine; see also FIG. 8, which shows, for example, a subroutine Y called by both the main routine and by a second routine X).

With respect to claim 11 (original), Alexander in view of Smolders further discloses the limitation wherein said collecting section stores said second profile and calling relationship information relating to said second profile, said calling relationship information indicating a relationship between said other subroutine and said called subroutine, in said storage unit (see Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including calling relationship information).

With respect to claim 13 (currently amended), Alexander in view of Smolders further discloses the limitation wherein said collecting section stores times of calling of said subroutine corresponding to said branch destination address as said profile in said storage unit (see Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a subroutine address and a base time; see also FIG. 4A, which shows timestamps for entering, i.e. times of calling, and returning from subroutines).

With respect to claim 14 (original), Alexander in view of Smolders further discloses the limitation wherein said collecting section obtains an overhead of said subroutine as said profile and stores said overhead in said storage unit (see Alexander, FIG. 5 and column 6, lines 37-53,

which shows the data structure used to store the profiles, including a time consumed by a thread executing a subroutine, which is considered a measure of overhead).

With respect to claim 15 (currently amended), Alexander in view of Smolders further discloses the limitation wherein said collecting section, when said identified result is said calling instruction, stores an identifier of an executor of said subroutine corresponding to said calling instruction and said branch destination address in said storage unit (see Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a subroutine address; note the parent node pointers, which identify the executors of subroutines).

With respect to claim 16 (currently amended), Alexander in view of Smolders further discloses the limitation wherein said collecting section, when said identified result is said calling instruction and said branch source address and said branch destination address are addresses of said subroutines, stores said branch source address and branch destination address as calling relationship information indicating a callings source subroutine and a calling destination subroutine is said storage unit (see Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a subroutine address and calling relationship information), and stores at least one of the cumulative execution time and the times of calling in said calling destination subroutine in the call source subroutine, as said profile corresponding to said calling relationship information, in said storage unit (see Alexander, column 6, lines 37-53, which shows that the profile comprises both a base time and a cumulative execution time).

With respect to claim 17 (currently amended), Alexander in view of Smolders further discloses a setting section setting an execution environment of a source of said interrupt so as to generate said interrupt when said branch instruction is executed during the execution of said program (see Smolders, column 3, lines 58-61, which shows setting the execution environment to generate a trace interrupt after every branch instruction).

With respect to claim 23 (currently amended), the computer readable medium recited in the claim is analogous to the apparatus recited in claim 12 (see Alexander and Smolders as applied to claim 12 above; see also Alexander, column 11, lines 6-17, which shows a computer readable medium).

With respect to claims 19-22 and 24-28 (currently amended), the limitations recited in the claims are analogous to those of claims 8-11 and 13-17, respectively (see Alexander and Smolders as applied to claims 8-11 and 13-17 above).

With respect to claim 34 (currently amended), the method recited in the claim is analogous to the apparatus recited in claim 12 (see Alexander and Smolders as applied to claim 12 above).

With respect to claim 30-33 and 35-39 (currently amended), the limitations recited in the claims are analogous to those of claims 8-11 and 13-17, respectively (see Alexander and Smolders as applied to claims 8-11 and 13-17 above).

With respect to claim 40 (new), Alexander in view of Smolders further discloses the limitation wherein the collecting section generates a control table corresponding to each executor

of the subroutine on the storage unit (see Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles; note that a path in the tree represents the subroutines executed by a specific executor; see also FIG. 9 and column 8, lines 24-33, which shows the data structure in the form of a table),

wherein the control table includes an executor managing table, a subroutine managing table, and a calling managing table (see Alexander, column 6, lines 54-61, which shows that the data structure may include other pointers and tables to aid in subsequent analysis).

wherein the executor managing table stores an identifier of the executor and a pointer to assign the subroutine managing table (see Alexander, FIG. 5 and column 6, lines 37-53, which shows parent node pointers that identify the executors of subroutines),

wherein the subroutine managing table is generated every subroutine executed by the executor, the subroutine managing table storing a subroutine address, times of calling of the subroutine, a cumulative execution time of the subroutine, the last called time of the subroutine, and a pointer to assign the calling managing table (see Alexander, FIG. 5 and column 6, lines 37-53, which shows a subroutine address, a base time and a cumulative time; see also FIG. 4A, which shows timestamps for entering and returning from subroutines), and

wherein the calling managing table is generated every subroutine called by the subroutine, the calling managing table storing a branch source address as a calling subroutine address, a branch destination address as a called subroutine address, times of calling of the called subroutine, a cumulative execution time of the called subroutine, the last called time of the called subroutine, and a pointer to specify the subroutine managing table managing the calling subroutine (see Alexander, FIG. 5 and column 6, lines 37-53, which shows a subroutine address,

a base time and a cumulative time, see also FIG. 4A, which shows timestamps for entering and returning from subroutines; see also column 5, lines 41-62, which shows a call or branch source address and a return or branch destination address for a subroutine).

Although Alexander in view of Smolders does not expressly disclose the recited table names and pointers, Alexander in view of Smolders discloses that the data structure may include such pointers and tables to aid in subsequent analysis (see Alexander, column 6, lines 54-61). Alexander in view of Smolders further discloses the recited information stored in the data structure, as presented above.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the data structure of Alexander in view of Smolders with other pointers and tables, as suggested by Alexander, including an executor managing table, a subroutine managing table, and a calling managing table, each with corresponding pointers, for the purpose of facilitating subsequent analysis of the profiles.

With respect to claims 41 (new), the limitations recited in the claim are analogous to those of claim 40 (see Alexander and Smolders as applied to claim 40 above).

With respect to claim 42 (new), the limitations recited in the claim are analogous to those of claim 40 (see Alexander and Smolders as applied to claim 40 above).

Conclusion

3. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

TUAN DAM
SUPERVISORY PATENT EXAMINER

MY

Michael J. Yigdall
Examiner
Art Unit 2122

mjy